RL-TR-95-260
In-House Report
Dec 1995

VHDL MODEL VERIFICATION AND ACCEPTANCE PROCEDURE

Michael P. Nassif and Mark T. Pronobis

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

Rome Laboratory
Air Force Materiel Command
Rome, New York

This report has been reviewed by the Rome Laboratory Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be releasable to the general public, including foreign nations.

RL-TR-95- 260 has been reviewed and is approved for publication.

APPROVED: Eugene C. Blackburn

EUGENE C. BLACKBURN
Chief, Electronics Reliability Division
Electromagnetics & Reliability Directorate

FOR THE COMMANDER: John J. Bart

JOHN J. BART
Chief Scientist
Electromagnetics & Reliability Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave Blank)

2. REPORT DATE
December 1995

3. REPORT TYPE AND DATES COVERED
In-House -- --

4. TITLE AND SUBTITLE
VHDL MODEL VERIFICATION AND ACCEPTANCE PROCEDURE

5. FUNDING NUMBERS
PE - 62702F
PR - 2338
TA - 01
WU - 8L

6. AUTHOR(S)
Michael P. Nassif and Mark T. Pronobis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Rome Laboratory/ERDD
525 Brooks Road
Rome, New York 13441-4505

8. PERFORMING ORGANIZATION REPORT NUMBER
RL-TR-95-260

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Rome Laboratory/ERDD
525 Brooks Road
Rome, New York 13441-4505

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Rome Laboratory Project Engineer: Michael P. Nassif (ERDD)/315 330-2241

12a. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)
The goal of this procedure is to establish a set of guidelines to be used by the DOD and industry for determining that a VHDL model is a functionally correct representation of the intended unit/device. In addition, this document provides VHDL model developers the means to evaluate their models against a set of criteria that the Government will be using in evaluation whether a VHDL model has captured the necessary design information. The verification procedure consists of six sections. The first section, Scope, provides an overview of the motivation behind the DOD requirement for VHDL models. This section also addresses models that will be archived and the minimal simulation environment needed for model evaluation. The second section, Referenced Documents, explains the order of precedence for reference documents in determining the functionality, timing and operation of the electronic device. Next is the Initial Inspection Section, which is a visual examination of the delivered files for proper documentation and format. The final three-sections, Detailed Inspection, Testing and Data Analysis, and the Final Report, includes a detailed examination and execution of VHDL source code, library components, header information and a report that discusses the findings.

14. SUBJECT TERMS
VHSIC Hardware Description Language (VHDL), hardware verification

15. NUMBER OF PAGES
36

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

1.0 SCOPE

The Department of Defense (DOD) is engaged in a number of programs which require VHDL (VHSIC Hardware Description Language) models of ASIC's and systems. Specifically, the details of the deliverable VHDL models are expressed in a combination of documents such as MIL-STD-454, the VHDL Data Item Description (VHDL-DID (DI-EGDS-80811)) and any additional requirements specified in any given Contract Deliverable Data Items ("CDRLs" or "data items").

VHDL data items capture the behavior and structure of an electronic system, subsystem, or device. The primary purpose of these data items is to document hardware designs in a machine executable, simulatable, and hierarchical format. VHDL models themselves must be inspected to insure that they meet the requirements specified in the contract or VHDL-DID, as applicable. The VHDL-DID may be tailored by the contract requirements for some applications.

For acceptance, VHDL simulation models provided to the Government as CDRLs must satisfy some known acceptance and verification criteria and procedure. These criteria and procedures are the purpose of this document.

The verification procedure includes model evaluation for compliance with the VHDL-DID, inspection and testing of the code for VHDL correctness, verification of models against the supplied WAVES (Waveform and Vector Exchange Specification)test vectors, verification of the models against the functionality of the described part, and verification of the model against the part specifications. Such verification methodologies will require an in-depth knowledge of VHDL simulation, electronics hardware functionality, and electronics test.

This document shall be used as the procedure document for the verification of VHDL simulation models supplied to the Government under contract, for certification and qualification under the new Qualified Manufacturers List (QML), or as part of Line Replaceable Module (LRM) acceptance.

2.0 REFERENCED DOCUMENTS

The first step in model verification is to obtain a set of specifications and references concerning the device or system being modeled. This information is then used by the individual performing this verification and acceptance procedure to educate themselves as to the functionality, timing and operation of the electronic system. Expert level understanding of the system's design, functionality and timing are essential prerequisites of the verifier.

2.1 Order of Precedence

The order in which the publications are listed below shall be the order of precedence in the event that one publication modifies the specifications or statements of a document at a higher level of precedence.(i.e. requirements under 3.3.2 override conflicting requirements at 3.3.3 and so on)

2.2 System Specifications:
Any or all of the foregoing specifications may contain block diagrams, timing charts, truth tables, stimulus response vectors, schematics and any additional information.

2.2.1 Standard IC DataBooks / Specifications:
The verifier shall obtain a copy of the commercially available device or system databook or specification if one is available from the manufacturer.

2.2.2 ASIC Design Specification:
For each ASIC undergoing verification and acceptance under this procedure, a detailed design specification shall be obtained. MIL-STD-454M (4.1.5)

2.2.3 System Level Specifications:
For systems incorporating more than one of any combination of ASICs or Standard ICs, a detailed system specification shall be obtained. MIL-STD-454M (4.1.5)

2.2.4 Hardware Test Plan:
For any system, subsystem, board or ASIC undergoing verification under this procedure, a detailed test plan shall be obtained for each design unit undergoing verification.

2.3 IEEE Publications
The following Institute of Electrical and Electronics Engineers (IEEE) publications are referenced either explicitly or implicitly within this document. The verifier should make each of these documents available to themselves for reference. Copies of the standards may be obtained from IEEE Standards Sales, 445 Hoes Lane, P.O. Box 1331, Piscataway, NJ 08855-1331.

2.3.1 IEEE Std 1076-1987: IEEE Standard VHDL Language Reference Manual (VHDL-LRM), 1988, The Institute of Electrical and Electronics Engineers, Inc. 345 East 47'th Street, New York, NY.

2.3.2 IEEE Std 1029.1-1992: IEEE Standard (Waveform and Vector Exchange Specification) Language Reference Manual, 1991, The Institute of Electrical and Electronics Engineers, Inc. 345 East 47'th Street, New York, NY.

2.4 Government Documents
The following US. Government standards are referenced either explicitly or implicitly within this document. The verifier shall make each of these documents available to themselves for reference. Copies of the standards may be obtained from the US. Government by contacting:

> Naval Publications and Printing Service Office
> 700 Robins Av.
> Philadelphia, PA. 19111-5094

The following illustration (Figure a) is a flow diagram of the verification procedure.

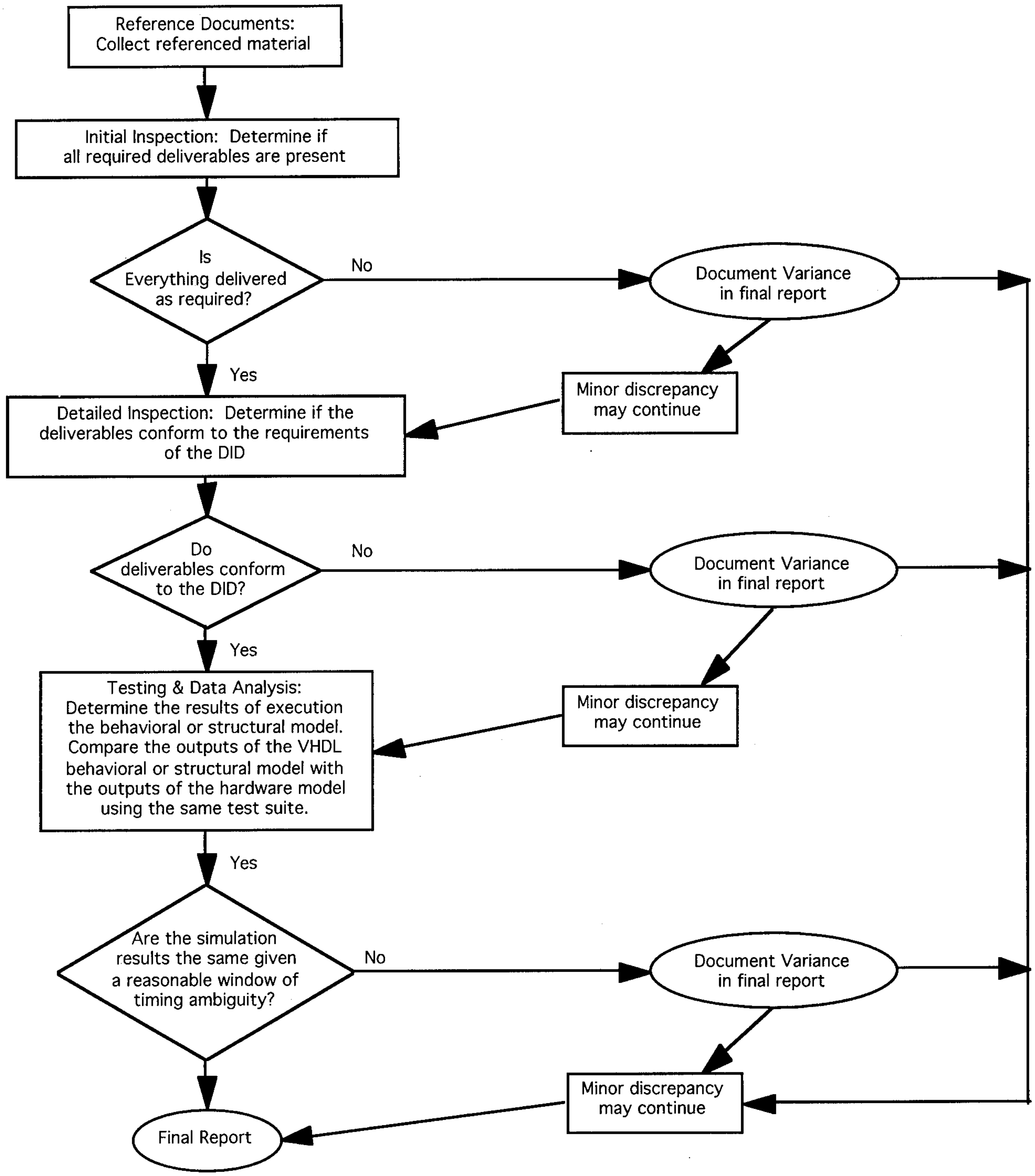

Figure a.

2.4.1 Military or Contract Specification:
The verifier shall obtain a copy of any applicable military or contract specification in addition to those mentioned above under which the VHDL Model(s) have been developed.

For some items, more than one specification (SMD, SCD, commercial, etc.), may be required, in which case it must be determined which specification the model is supposed to represent. Also, many specifications may differ by timing alone so that one model with different timing packages may satisfy several different military specifications (or "dash numbers"). In addition, the same part may occur in different packages.

2.4.2 MIL-STD-454M, Requirement 64 published April 1991

2.4.3 DI-EGDS-80811 VHSIC Hardware Description Language (VHDL) Data Item Description.

2.5 Verification Procedure

To facilitate the performance of this verification procedure, the verifier shall follow the instructions provided in section 6.0.

Note 1: Hereafter MODEL shall refer to the VHDL code delivered to represent a digital electronic unit/device/component. The MODEL may be described as a high level (behavioral) model or as a gate level (structural) model or as a combination of behavioral and structural (mixed) model.

Note 2: The intended unit/device/component for which the MODEL was developed will hereafter be called the REFERENCE. The REFERENCE item may be hardware, or if no hardware exists it will be the specifications for the intended unit/device/component.

3.0 INITIAL INSPECTION

3.1 Documentation Files Required under DID DI-EGDS-80811

The verifier shall determine that the contractor has provided all of the system specifications, hardware test plans, and any additional documentation required for the verifier to determine the functionality and timing of the system, subsystem or device undergoing verification.

This section explains which items are to be "visually" inspected to determine whether all the required deliverables are present, in the proper order and that they meet certain criteria specified in the VHDL-DID.

The following eight auxiliary information files shall precede VHDL design files.

3.1.1 Table of Contents File:

Inspect that this file contains the names of each of the VHDL files delivered; one file name per record and nothing else (pad with trailing blanks).
(DID requirement 10.3a)

File should be an ASCII file. Comments should begin with the character #. ONLY one file per line.

Note 3: From a model style guide perspective, the developers should be encouraged to deliver the models in the following format: <package>.vhd for package declarations, <package>_body.vhd for the corresponding package body, <modelname>_e.vhd for model entities, <modelname>_a_str.vhd for structural architectures, <modelname>_a_beh.vhd for behavioral architectures, <modelname>_c_str.vhd for structural configurations, <modelname>_c_beh.vhd for behavioral configurations, and <modelname>_test bench.vhd for test benches. This information is provided as a guide, not a requirement. The developer is encouraged to follow a similar file naming methodology.

Note 4: In addition, it is suggested that each model's entity, architectures, test benches and configurations be located in its own sub-directory under the name of the model. The same holds true for packages. In addition, all simulation scripts and results can be located in the same sub-directories as the models to which they pertain.

3.1.2 CDRL File:
Inspect that this file contains a high-level prose description of the VHDL deliverables. The description shall contain the following: (a) contract, (b) line item, (c) Contract Data Requirements list sequence number and in addition, (d) summarizing the organization and content of the set of files.(DID requirement 10.3b)

3.1.3 Analysis File:
Inspect that this file contains a specification of the order of analysis of the VHDL design units. Verify that the order of analysis is consistent with the rules of VHDL.
(DID requirement 10.3c)

(i.e. Packages compiled before Package bodies, which in turn are compiled before the Entities/Architectures or Configurations which reference them).

3.1.4 Leaves File:
Inspect that this file contains the list of unmodified VHDL leaf-level models, that have been provided by the Government, which have been referenced within any of the VHDL files mentioned in.
(DID requirement 10.3c)

3.1.5 Modifications File:
Inspect that this file contains the list of modules previously accepted by the Government.
(DID requirement 10.3e)

3.1.6 Deliverables Files:
Inspect that this file contains a list of VHDL modules which originate with this VHDL delivery.
(DID requirement 10.3f)

3.1.7 Test bench Association File:
Inspect that this file contains a list which associates VHDL modules with their corresponding test benches.
(DID requirement 10.3g)

3.1.8 Auxiliary Information File(s):
Inspect that this file(s) contains any additional information concerning the VHDL descriptions and VHDL design files. Inspect that the contents of the auxiliary files do not contain any complete VHDL Design Units.
(DID requirement 10.3h)

3.2 Conformance to IEEE VHDL-1076
The verifier is instructed to compile (analyze) the VHDL files, on a fully compliant VHDL IEEE-1076 analyzer, in the order specified in the Analysis File delivered under SECTION 3.1.3. Each of the files shall analyze with no errors. Certain analyzers will issue warnings. The verifier shall make a record of the execution of the analyzer and specifically note any errors or warnings indicated.

4.0 DETAILED INSPECTION

The second phase of the verification process is a detailed inspection of entities, architectures, configurations and other support modules delivered.

4.1 Comment Banner:
In order to assist the model verifier, a comment section is required to precede each VHDL module. The format of the comment section should be similar to:

*Design unit name identifier:
*Identification of originator or source:
DOD approved identifier (if one exists):
*Whether model has been previously delivered:
*General approaches taken to modeling, and particular
 decisions regarding Modeling fidelity:
*Any further information vital to subsequent users of the
 descriptions:
*Any factors restricting the general use of this
 description to represent the actual hardware:
*Any assumptions taken in developing the model:
*Was the module previously approved by the DOD?

4.1.1 Comment Banner with Revision Information:
If the module is a previously approved module and has been revised with this delivery, the following information shall also be included.

*The date of revisions:
*The performing individual and organization:
*The rationale for the revision:
*A description of which part of the original design unit
 which required modification:
*A description of the testing done to verify the revised
 model:

4.1.2 Comments:
While it is difficult to determine quantitatively that the model author has sufficiently commented the VHDL code, a usual rule of thumb is to have an approximate 20% comment overhead.

4.1.3 Inspection for Orthogonality:
The files shall be inspected to ensure that each file is either a VHDL design file, whose entire contents conform to the requirements of the VHDL Language Reference Manual, or an auxiliary information file containing no VHDL design units.
(DID requirement 10.3)

4.1.4 Inspection for Incremental Information:
The files shall be inspected to ensure that new design units are not contained in the same file as design units which have been previously accepted by the Government.
(DID requirement 10.3)

Note 5: Insuring that a previously accepted module has not been altered may be performed using a text comparison to discover any differences between the archived module and the delivered module. Differences other than variable names and comments should be examined for their effect on module functionality, these differences should be noted in the final report.

4.2 Model Evaluation and Inspection
The procedure described in this section should apply to Entities, Architectures, and Configurations of the Model. All material in section 4.1 pertains to each module of the VHDL deliverables.

4.2.1 Entity Declaration (DID Conformance):
Each entity declaration shall be inspected for the specifications listed in this section. In addition, if the Entity is contained in a separate file, then the procedures pertaining to revision information and comments, (section 4.1), shall apply as well.

4.2.1.1 Entity Declaration:
The entity declaration for each entity shall include:

a) An interface declaration
b) Timing and electrical requirements for the behavior of the device
c) Allowable operating conditions
d) Component identification
e) Explanatory comments.

(DID requirement 10.2.2)

4.2.1.2 Entity Interface Declaration:
The interface declaration for each entity shall be inspected to assure:

a) that a description has been included for every port that exists on the device.

b) the inclusion of information relating each input and output port to a package pin number or connector pin number whenever such a correspondence exits. (DID requirement 10.2.2.1)

Note 6: If a condition should arise such that the name of the port violates the rules of VHDL, an appropriate alternative name should have been selected and commented as such.

Note 7: There are a number of ways in which this information may be obtained including (a) comments, (b) port attributes, or even the instantiation of a "packaging" entity whose port names correspond to the pin numbers of the packaging of the device (i.e. Pin_23 as a port name). (DID requirement 10.2.2.1)

4.2.1.3 Entity Naming Conventions:
The entity declaration shall be inspected to ensure that the names for VHDL entities are traceable to the names of their physical electronic counterparts whenever such a correlation exists.
(DID requirement 10.2.2.4)

4.2.1.4 Timing & Electrical Requirements:
The model shall be inspected to ensure that timing and electrical requirements are expressed in such a manner as to cause the simulator to generate error messages upon violation of a specification during simulation.
(DID requirement 10.2.2.2)

The specifications may include the following:

- Timing specifications such as Setup, Hold, Pulse Width, Periodicity, and Release or Recovery times, among others.

- Electrical Specifications such as maximum fanout DC load, maximum fanout capacitive load, maximum drive current limits, Voltage range, Temperature range, etc.

- Additional timing considerations such as required number of clock cycles for correct reset to occur, etc.

4.3 Architectures
Each architecture declaration shall be inspected for the specifications listed in this section. In addition, if the architecture is contained in a separate file, then the procedures pertaining to revision information and comments shall apply as well.

4.3.1 Hierarchy:
Inspect that the models delivered are written with a "reasonable" level of hierarchy. The model shall be inspected to ensure that structural decomposition of behavioral bodies is used only when necessary to show functional partitions of the corresponding structural body. Ease of simulation and clarity of behavior shall be considered when determining the appropriate level of hierarchical decomposition.
(DID requirement 10.2.3.1)

4.3.1.1 The model shall be inspected to ensure that the hierarchy of VHDL modules is analogous to the physical hierarchy of the hardware being documented. The model shall be inspected to ensure that one VHDL module is defined for the entire system, and one for each physical electronic unit (assembly, subassembly, integrated circuit, etc.) of the hardware system, and that VHDL modules are defined for important subsections or groupings of complex physical units (e.g., macrocells of a chip or boards defining a processor).
(DID requirement 10.2.1)

Note 8: As a guide, an ASIC should have a minimum of three (3) levels of hierarchy: (a) A behavioral model of the ASIC at the pin boundary level (no structural sub-architectures), (b) a level representing the ASIC's block diagram (which includes structural sub-architectures), where the structural sub-architectures are written as behavioral models and lastly (c) a detailed gate-level architecture where all of the components are leaf-level models.

4.3.2 Physical Correspondence:
Inspect that the model's architecture is written and commented sufficiently well such that the internal signal names and hierarchical component names reasonably match the names of the physical implementation.
(DID requirement 10.2)

4.3.3 Signal delays:
Inspect that all signal delays accurately model the behavior of the device specification. At a minimum, the models shall be coded to incorporate a means of evaluating minimum, typical, and maximum timing delays. More elaborate timing models which take into account other variables such as supply voltage or output loading may also be used.
(DID requirement 10.2.3.2)

Note 9: Determining that the model "accurately" models the timing of a specification is a difficult task. Certain areas to look for include:

- All possible input to output pin asynchronous "cause-effect" paths have a corresponding delay path. This delay path may, in addition, be level sensitive.

- Determine how the model should respond under conditions when simultaneous events could trigger events which preempt previous timing events causing the output to change to a new state at the wrong time.

- Normally, inertial delay should be used. However, certain conditions, such as glitch detection, require a transport delay mechanism.

4.4 Behavioral SubArchitecture
Each behavioral subarchitecture shall be inspected to ensure that it meets the specifications listed in this section.

4.4.1 Visibility of Internal registers:
The model shall be inspected to ensure that all user programmable operations and registers are clearly identifiable in the simulation model. The model verifier shall make a check list of the programmable operations and registers for later use.

(DID requirement 10.2.3)

4.4.2 Test and Maintenance Functions:
Inspect that, if test or maintenance functions are available to the user of the actual component, the model includes a description of the test functions.
(DID requirement 10.2.3)

4.4.2.1 Test and Maintenance Functions for Behavioral Models:
Detailed structural scan signature paths shall not be specified. However, the device interface declaration (Entity port list) should include the scan test port declarations. The model shall be inspected to ensure that signal values which are dependent on a particular structural implementation, such as scan path signatures, are not specified in the behavioral body.
(DID requirement 10.2.3.3)

Note 10: In addition, the behavioral model, when placed into a test mode, should respond with a NOTE level assertion stating that the scan structure has not been implemented in the model.

4.5 Structural SubArchitecture
Each structural subarchitecture shall be inspected to ensure that it meets the specifications listed in this section.

4.5.1 Test and Maintenance Functions:
Inspect that, if test or maintenance functions are available to the user of the actual component, the model includes a description of the test functions.
(DID requirement 10.2.3)

4.5.2 Test and Maintenance Functions for structural models:
The model shall be inspected to ensure that structure which is created to support testing and maintenance (such as scan path signatures) is included in the VHDL structural description.
(DID requirement 10.2.4)

Detailed structural scan signature paths shall be specified.

4.5.3 Correspondence to Actual Implementation:
The model shall be inspected to ensure that the structural bodies represent the physical implementation. The details of the model at this level should enable logic fault modeling and test vector generation to be performed, not necessarily within a VHDL environment
(DID requirement 10.2.4)

4.5.4 Traceability:

The model shall be inspected to ensure that the names of components and signals are the same as, or traceable to, their electrical schematic counterparts, for ease of schematic drawing correlation, and within the constraints of the lexical rules of VHDL. (DID requirement 10.2.4.1)

4.5.5 Leaf-Level Modules:
The model shall be inspected to ensure that each leaf level module can be classified in one of the following categories:

- Modules selected from a Government list of leaf level modules.

- Modules corresponding to a collection of hardware elements which together exhibit a stimulus-response behavior, but whose interaction is best modeled at the electrical or physical level. Examples of such modules are digital logic gates, analog circuit blocks, and power supplies.

- Modules whose detailed design has not yet been completed, but whose behavior is required as an interim contractual deliverable.(DID requirement 10.2.1.1)

4.6 Dataflow SubArchitecture
Each of the procedures defined in and above shall be applied to dataflow modeling subarchitectures as well.

4.7 Inclusion of Packages
The model shall be inspected to ensure that VHDL package declarations are used whenever operating conditions are common across a class of similar components. (DID requirement 10.2.2.3).

Note 11: Operating conditions are the physical and electronic environment in which components are designed to operate, such as temperature range, signal excursions, logic level definitions, maximum power dissipation, radiation hardness, etc.

4.7.1 Traceability:
Inspect that all such specifications are traceable back to the physical device specifications. (DID requirement 10.2.2.3)

4.8 Test benches
4.8.1 Check for Existence of Corresponding Test bench:
Every VHDL module shall be simulatable as a stand alone module and hence a corresponding VHDL test bench is required for every VHDL module of the hierarchy. (DID requirement 10.2.5.3)

4.8.2 WAVES Conformance Requirements:
The test vectors shall be inspected to determine that they have been written in the WAVES format.

4.8.3 Distinguishable from the module:

The test benches shall be inspected to ensure that they are clearly distinguishable from the VHDL modules representing the design itself.
(DID requirement 10.2.5)

4.8.4 Test bench Comments:
The test bench shall be inspected for explanatory comments. Refer to section 4.1 and 4.1.2.
(DID requirement 10.2.7)

4.8.5 Test Vector(s) description:
A detailed description of the purpose of each test bench shall be included.
(DID requirement 10.2.7)

4.8.6 Assertion Reports:
The test bench shall be inspected to assure that it stimulates the Module Under Test and reports any discrepancies with expected response during simulation.
(DID requirement 10.2.5.1)

4.8.6.1 Assertion Messages:
For any error messages, inspect that they identify the requirement that has failed, and that the error message includes the name of the violating VHDL design entity.
(DID requirement 10.2.6)

4.8.7 Sufficiency of Configuration Information:
Inspect the test bench to assure that sufficient configuration information is present to facilitate the test.

4.8.8 Test Requirements Correlation:
The VHDL test benches and the hardware test drawings and test plans shall be inspected to ensure that they are cross-referenced to any required hardware test plans as necessary.
(DID requirement 10.2.5.2)

4.8.9 Necessary Tests:
Each test bench shall be inspected to ensure that the WAVES test vectors used within it are necessary to simulate the correct behavior of the VHDL module to which it corresponds.
(DID requirement 10.2.5)

4.8.10 Sufficient Tests:
Each test bench shall be inspected to ensure that the WAVES test vectors used within it are sufficient to simulate the correct behavior of the VHDL module to which it corresponds. This includes a sufficient set of test vectors to violate all timing constraints.
(DID requirement 10.2.5)

4.9 Configurations

Note 12: While there are no specific procedures to verify configurations, a number of issues should be pointed out as follows:

Default Bindings: When default bindings are used, a comment stating such would be useful.

OPEN Associations: When OPEN ports or Design Units are encountered in the configuration, a comment should be made as to the purpose of the OPEN association.

Note 13: Type Conversions: When type conversion functions are used to pass signal data from one VHDL model to another, a comment should state if the mapping is identical or not. If the mapping is not identical, then the comment should state whether any unmapped signal values are likely and to which state they are being mapped.

5.0 Testing and Data Analysis

Verifying the correct behavior of a VHDL simulation model is a complex undertaking. The verifier needs a detailed understanding of the device that has been modeled. All sources of information concerning the device's operation shall be obtained and used to determine what testing is required to verify that the delivered VHDL simulation model operates correctly and if the delivered test suite is sufficient to meet those requirements. The intent of the verification is to detect errors or omissions in the functionality, timing, style or content of the VHDL simulation model. The mechanics of the verification procedure are dependent on the amount of design detail available for the REFERENCE and the amount of design detail available for the MODEL.

Simulation results must match those indicated by the specification for items such as best/worst/nominal output delays versus temperature and voltage ranges. Error messages caused by timing violations shall be inspected to ensure that they correctly identify the requirement which has been violated and the name of the VHDL design unit in which the error has occurred.

The testing procedure consists of:

a) Comparing the operation of the MODEL to the specifications of the REFERENCE.
b) Comparing the operation of the MODEL to the operation of the intended REFERENCE

5.1 Definitions for Testing:

5.1.1 Test Bench: A VHDL module which apply WAVES stimuli to a unit under test (UUT), compare the UUT's response and the WAVES expected output, and report any differences between observed and expected responses during simulation. Each configuration should have a corresponding test bench which is clearly distinguishable from the UUT modules.

5.1.2 Test Suite: A collection of one or more test benches to which is associated a corresponding MODEL.

5.1.3 Test bench Configuration Sets: Determine, that a test bench configuration set has been made available for every pair of entity/architecture/test benches such that a test bench configuration exists for each pair. (i.e.. MODEL-structural_view + Test 1, MODEL-behavioral_view + Test 1, etc....).
(DID requirement 10.2.5.1)

5.1.4 Commercial Model REFERENCE: For commercially available integrated circuits. The DID specifies that the VHDL test case shall correspond to an equivalent hardware test plan. If no such test plan exists, as in the case of a model of a standard IC device, then the REFERENCE shall be the actual device.

5.2 Execution of Test Suite:
Each test bench shall be executed and the results of the simulation runs recorded.
(DID requirement 10.2.5.1)

5.2.1 Behavioral and Structural Verification:
Run every test bench configuration set and record the results.
(DID requirement 10.2.5.1)

5.2.2 Automatic Checking:
The simulation results shall be analyzed to ensure that each VHDL test bench does correctly applies stimuli to the MODEL, compares the MODEL's response to an expected WAVES output, and reports any differences between observed and expected responses during simulation.
(DID requirement 10.2.5.1)

This involves monitoring the test bench and the MODEL during simulation to insure that the proper functions in the MODEL are activated by the test bench and that the proper responses occur in the MODEL and are properly monitored by the test bench. The comments provided with the test bench defines what should happen with each test bench.

5.2.4 Augmentation of test vectors:
The model developer shall provide WAVES test vectors designed to check functionality and timing with a comment provided for each vector or set of vectors describing the associated function being tested. The model verifier shall develop a set of test vectors which violate the timing and voltage specifications, attempt to perform illegal model operations and test its functionality at its operational limits if not provided by the developer. Any additional vectors developed to augment the test vector set shall be documented in the final report. The UUT shall then be simulated and the results analyzed.

5.2.3 Determination of REFERENCE Test Goodness:
Referring back to the REFERENCE specifications and the hardware test plan:

5.2.5 REFERENCE Test Coverage Determination: Determine if there exists any REFERENCE behavior specified in the functional specification that is not tested by a test bench. This involves comparing the functional specification with the test bench descriptions to identify test bench omissions.
(DID requirement 10.2.5.3).

5.2.9 Augmentation of Test bench(s): If in the opinion of the verifier, additional test bench(s) are warranted, then the verifier may write those test benches and document the purpose of each test.

5.2.10 Simulation: The VHDL modules shall be simulated on any available IEEE-1076 compliant simulator using the supplied test vector set in the WAVES format.

5.3 Results Analysis
5.3.1 Result Documentation:
Document every test performed under this section. Note any errors or omissions and write additional test benches as deemed necessary.

5.3.2 MODEL Functionality Omissions:
Look for REFERENCE behavior specified in the functional specification that is not modeled at all in the MODEL. This involves comparing the functional specification with the MODEL to identify MODEL omissions.

5.3.3 MODEL Functionality Errors:
Look for REFERENCE behavior specified in the functional specification that has been modeled in error in the MODEL. This involves comparing the functional specification with the MODEL to identify MODEL errors. REFERENCE behavior includes timing behavior and functional behavior.

5.3.4 MODEL Timing Performance:
Check for proper modeling and testing of best, worst and nominal output delays.
(DID requirement 10.2.2.2).

5.3.4.1 Among other timing test situations to look for, the following is a list of timing conditions commonly found in commercial device models:

- setup, hold, recovery, and release time specifications
- Periodicity, pulse-width and cycle-count specifications
- timing variations due to voltage, temperature or loading
(DID requirement 10.2.3.2).

Note 14: Performing this procedure involves monitoring the MODEL during simulation to insure that the proper timing relationships exist in the MODEL and that they are activated by the test bench. The comments provided with the test bench defines what should happen with each test bench.

5.3.5 Timing Violation Error Reports:

The error messages caused by the timing violation shall be inspected to ensure that they correctly identify the requirement which has been violated and the name of the VHDL design unit in which the error occurred. Applicable VHDL design units include: entity declarations, architectures, package declarations, package bodies, and configurations. (DID requirement 10.2.6).

OPEN ISSUE: An alternative to verifying timing with simulation is to verify timing with a static timing analysis tool.

5.3.6 MODEL programmable operations performance:
Check for proper operation of all user programmable operations (instructions, registers, etc.)
(DID requirement 10.2.3).

Note 15: This involves comparing the REFERENCE specification with the MODEL to identify MODEL errors or omissions. In addition, common areas to investigate include instruction operation in all addressing modes, explicit use of illegal opcodes, and determination that instructions execute in proper time sequence with the correct cycle count among other test.

5.3.7 MODEL Test and Maintenance Functions:
Check for proper operation of all test and maintenance functions that are available to the user.(DID requirement 10.2.3)

5.4 REFERENCE Implemented in Hardware
A correlation between the actual hardware and the VHDL model to ensure correctness is the next step of the testing process. The same WAVES test vectors used to stimulate the MODEL will be used to test the corresponding hardware. At this level, discrepancies would indicate a failure in the MODEL's description, an incorrect test vector set, or hardware that fails to meet the specification.

This procedure shall be applied when the actual hardware component is considered to be of higher quality than the VHDL model. This is normally the case whenever a third-party develops a behavioral model of a commercial digital integrated circuit from the description contained in a non-proprietary databook or data sheet.

5.4.1 Hardware Test Fixture:
Construct or mount the REFERENCE into a hardware test fixture (for a commercial component, this is usually a hardware modeler interfaced to a digital simulator). Develop a means of applying the test patterns generated by the Test Suite to the REFERENCE. (In a typical VHDL-based simulator with a hardware modeler interface, this requires the writing of a configuration design unit binding the formal component instantiation to the physical device through the hardware modeler software interface protocol).

5.4.2 Verification Procedure:
Repeat sections 5.2 through section 5.3.7 with the physical REFERENCE.

5.4.3 Test Response Comparisons:
Compare the responses of the REFERENCE against the response of the MODEL.

5.4.3.1 Comparison Considerations: The intent of comparing the responses of the MODEL with the responses of a REFERENCE is to insure the MODEL reflects the behavior of the REFERENCE both functionally and to some allowable timing tolerance. Because of the many possible differences that may exist between the two, special care needs to be taken to insure a valid comparison.

5.4.4 Different Levels of Abstraction
By itself, different levels of modeling abstraction between the REFERENCE and the MODEL shouldn't present additional problems. The REFERENCE operation may be encapsulated in a written specification, a high level (behavioral) model, a gate level model or by the hardware; the MODEL may be described as a high level (behavioral) model or as a gate level model. But the DID only mandates that physical I/O pins, timing characteristics on I/O pins and user accessible hardware objects be clearly identifiable and hence comparable. Other internal objects may or may not match across the two models and certainly should not be used as a basis for comparison.

5.4.5 Data Sampling
5.4.5.1 While there will be differences in the details, both models are intended to represent a common behavior. Keeping in mind the issues presented below, a reasonable comparison is possible.

5.4.5.2 The comparison is only as good as the data being compared. Getting good data is a function of proper sampling. Proper sampling is determined by the amount of timing detail incorporated into the REFERENCE and the MODEL. Sampling rates, times and locations are determined by the REFERENCE or MODEL with the least accurate timing for pin to pin comparisons. For internal comparisons, sampling rates, times and locations are determined by the REFERENCE or MODEL with the most abstract description.

5.4.6 Strobing Intervals / Time Offset:
If both models contain accurate timing information then sampling can occur at regular timed intervals, for instance: every 5ns. This interval is determined by the level of accuracy required in the comparison and the allowable timing tolerances. Data collection must be offset far enough from sample initiation to guarantee valid data in the presence of any modeled or physical delays and any timing ambiguity due to timing tolerances.

If one of the models does not contain detailed timing information, then sampling must be initiated with system clocks (synchronous design) or control signals (asynchronous designs). Data collection must be offset far enough from sample initiation to guarantee valid data in the presence of any modeled or physical delays and any timing ambiguity due to timing tolerances.

If both models contain accurate functional information then sampling can also occur at any internal location, for instance: at every register. These locations are determined by the level of accuracy required in the comparison. If one of the models does not contain detailed functional information, then sampling is only useful where there are common objects.

5.4.7 Cycle Count: Certain processors require n-clock cycles in order to perform a given function (i.e.. multiplication in 154 clocks on a MC68000). Check that the MODEL matches the REFERENCE with respect to cycle counts.

5.4.8 Timing Tolerance Windows:
Certain REFERENCE specifications indicate that the MODEL shall respond to a stimulus within a certain relative time interval with respect to the stimulus or a gating clock signal. Check that the test bench has been written in such a way as to determine that the response transition occurs within that timing window and that the test bench issues an error if the response (a) fails to occur, (b) fails to provide the correct value during the time window, or (c) occurs outside of the time window.

5.4.9 Discrepancies:
Document any errors or omissions and write additional test benches as necessary.

5.4.10 Justifiable Discrepancies:
Make a list of justifiable discrepancies indicating the discrepancy along with an explanation of why the discrepancy is acceptable.

5.5 Simulation Values:
If the REFERENCE and the MODEL have different value systems, a mapping from one to the other must be defined. This mapping will be used during the comparison process to insure response equality.

Consider the case where the REFERENCE uses a three state ('U', '0', '1'), two strength ('W', 'S') value system and the MODEL uses a five value system ('U', '0', '1', 'Z', '-'). The mapping might be something like ('W','U') -> 'U', ('S','U') -> 'U', ('W','0') -> 'Z', ('S','0') -> '0', ('W','1') -> 'Z', ('S','1') -> '1'. There is no mapping for the '-'.

Keeping in mind the issues of data sampling below, the comparison procedure uses the mapping defined above to determine when responses in the REFERENCE are equivalent to responses in the MODEL. If the MODEL ever exhibits a '-' during the comparison process, special analysis is probably called for to determine what is happening. The existence of the '-' doesn't automatically imply model differences.

5.6 Model Initialization
5.6.1 Here's a situation where it may appear that things can be more complicated than is actually the case. Because the models are supposed to be equivalent, the external stimulus that initializes the REFERENCE and the MODEL have to be identical. After that stimulus has been applied, the two models must be in identical states or the models aren't equivalent. But what constitutes identical states?

5.6.2 Consider the case where an initialization sequence consists of holding a reset line active and then applying 5 clock pulses. Upon completion of the initialization sequence, all state machines should be at their starting state, all registers should be cleared and all output buses should be tri-stated. A comparison of the two models is constrained in both space and time because of the way the specification is defined and because of what the DID allows.

5.6.3 The state machines in the two models cannot be observed and compared at all because they probably don't represent user programmable registers. The DID does not require that internal hardware objects be modeled to any standard or that they must behave in any set way. Only the subset of registers that are "user programmable" may be observed and compared for the same reason. Nothing in either of the models (registers, buses, etc.) may be compared during the reset sequence, only after it is completed. The specification and the DID make no statement about the behavior of the circuit or the models during the reset sequence. Because of the data sampling issues below, there may need to be some additional delay before the comparison sample is taken because of pin switching delays.

5.7 Unjustifiable Discrepancies:
Make a list of unjustifiable discrepancies indicating the discrepancy along with an explanation of why the discrepancy is unacceptable.

5.8 I/O Pin differences:
There are two possibilities here. One possibility is that a pin is present in one model and absent in the other. For instance: a high level model has no scan out pin because that behavior wasn't modeled. An "equivalent" low level model has a scan out pin. The other possibility is that a pin is present in both models but behaves differently due to "level of abstraction" issues. While it may seem there are extenuating circumstance here, there really aren't. In neither case are these models equivalent. The DID requires pin for pin compatibility.

6.0 Verification Report

A final report shall be written detailing the results of this Model Verification Procedure. The report shall contain the following sections.

6.1 Contents and Organization of the Report:

6.1.1 Final Report Header:
Contains essential information regarding the hardware being modeled and the modeling environment.
(see Appendix A)

6.1.2 Verification Procedure Checklist:
Assures that the model has been inspected against each item of the procedure.
(see Appendix B)

6.1.3 Final Report Format:
Explains the expected deliverable format of the final report.
(see Appendix C)

Appendix A

(Final Report Header)

Report Name :

Verificator Name :

Model Name :

Model Version :

Model Vendor :

Authorizing Requester :

Analyzer Vendor Name :

Analyzer Model :

Analyzer Version :

Simulator Vendor Name :

Simulator Version :

Hardware Modeler Vendor :

Hardware Modeler Model :

Hardware Modeler Version :

Source of REFERENCE data:

List of additional HW / SW used for this test:

List of auxiliary test benches:

Instructions to the Verificator :

Appendix B

(Verification Procedure Checklist)

The Verificator shall check that each task item has been accomplished as described in the Verification Procedure.

2.0 REFERENCED DOCUMENTS
- 2.2 System Specifications
  - 2.2.1 Standard IC DataBooks / Specifications
  - 2.2.2 ASIC Design Specification
  - 2.2.3 System Level Specifications
  - 2.2.4 Hardware Test Plan
- 2.3 IEEE Publications
  - 2.3.1 IEEE Standard VHDL Language Reference Manual (VHDL-LRM)
  - 2.3.2 IEEE Standard VHDL View of Waves (Waveform and Vector Exchange Specification)
- 2.4 Government Documents
  - 2.4.1 Military or Contract Specification
  - 2.4.3 DI-EGDS-80811 VHSIC Hardware Description Language (VHDL) Data Item Description.
  - 2.4.4 TISSS Specification

3.0 INITIAL INSPECTION
- 3.1 Documentation Files Required under DID DI-EGDS-80811
  - 3.1.1 Table of Contents File
  - 3.1.2 CDRL File
  - 3.1.3 Analysis File
  - 3.1.4 Leaves File
  - 3.1.5 Modifications File
  - 3.1.6 Deliverables Files
  - 3.1.7 Test bench Association File
  - 3.1.8 Auxiliary Information File(s)
- 3.2 Conformance to IEEE VHDL-1076

4.0 DETAILED INSPECTION
- 4.1 Comment Banner
  - 4.1.1 Comment Banner with Revision Information
  - 4.1.2 Comments
  - 4.1.3 Inspection for Orthogonality
  - 4.1.4 Inspection for Incremental Information
- 4.2 Model Evaluation and Inspection
  - 4.2.1.1 Entity Declaration
  - 4.2.1.2 Entity Interface Declaration
  - 4.2.1.3 Entity Naming Conventions
  - 4.2.1.4 Timing & Electrical Requirements
- 4.3 Architectures
  - 4.3.1 Hierarchy
  - 4.3.2 Physical Correspondence
  - 4.3.3 Single delays
- 4.4 Behavioral SubArchitecture
  - 4.4.1 Visibility of Internal registers

5.5 Simulation Values
5.6 Model Initialization
5.7 Unjustifiable Discrepancies
5.8 I/O Pin differences

Appendix C

Delivery of the Final Report : The Verification report, along with the Verification checklist shall be filed in ASCII version and be appended to the final tape provided for Acceptance. In addition, a written copy shall be provided to the Program Office requiring the Acceptance.

Certification of Verification

THIS VERIFICATION PROCEDURE has hereby been performed in accordance with the Verification Procedure attached hereto.

WHEREAS, Verificator has hereby certified that the Model(s) under consideration have been evaluated in accordance with the Verification procedures set forth in the Verification Procedure document; and

WHEREAS, the Verificator hereby represents that any discrepancies found have been indicated in an accompanying Verification Report attached to this Checklist; and

NOW THEREFORE, the Verificator certifies that such tests were performed as required by affixing his/her signature below.

Verificator Signature :

Date :

REFERENCES

[1] IEEE Std 1076-1987 : IEEE Standard VHDL Language Reference Manual (VHDL-LRM), 1988,

[2] VHDL View of Waves (Waveform and Vector Exchange Specification) Document, IEEE 1029.1/D2

[3] MIL-STD 454M, Requirement 64, 1991

[4] DI-EGDS -80811 VHSIC Hardware Description Language

*U.S. GOVERNMENT PRINTING OFFICE: 1996-710-126-20088

MISSION

OF

ROME LABORATORY

Mission. The mission of Rome Laboratory is to advance the science and technologies of command, control, communications and intelligence and to transition them into systems to meet customer needs. To achieve this, Rome Lab:

a. Conducts vigorous research, development and test programs in all applicable technologies;

b. Transitions technology to current and future systems to improve operational capability, readiness, and supportability;

c. Provides a full range of technical support to Air Force Materiel Command product centers and other Air Force organizations;

d. Promotes transfer of technology to the private sector;

e. Maintains leading edge technological expertise in the areas of surveillance, communications, command and control, intelligence, reliability science, electro-magnetic technology, photonics, signal processing, and computational science.

The thrust areas of technical competence include: Surveillance, Communications, Command and Control, Intelligence, Signal Processing, Computer Science and Technology, Electromagnetic Technology, Photonics and Reliability Sciences.